

I. REAL PARTY IN INTEREST

The subject application is owned by National Instruments Corporation, a corporation organized and existing under and by virtue of the laws of the State of Delaware, and having its principal place of business at 11500 N. MoPac Expressway, Bldg. B, Austin, Texas 78759-3504.

II. RELATED APPEALS AND INTERFERENCES

No related appeals or interferences are known which would directly affect or be directly affected by or have a bearing on the Board's decision in this appeal.

III. STATUS OF CLAIMS

Claims 2-8, 10, 14-19, 25 and 28-30 are canceled.

Claims 1, 9, 11-13, 20-24, 26, 27 and 31-34 are pending in the application. All of the pending claims stand rejected and are the subject of this appeal. A copy of the claims incorporating entered amendments is included in the Claims Appendix below.

IV. STATUS OF AMENDMENTS

All amendments have been entered. The Claims Appendix hereto reflects the current state of the claims.

V. SUMMARY OF THE INDEPENDENT CLAIMS

Claim 1 recites a computer-readable memory medium (*Block 166 of FIG. 3; p. 8, lines 15-27*) storing program instructions. The program instructions are executable to: in source code of a software program, display a first function call written in a text-based programming language that can be compiled into executable code. The first function call takes a first parameter. (*See block 301 of FIG. 4; FIG. 14; p. 22, lines 17-18*).

The program instructions are further executable to programmatically determine one or more valid parameter values for the first parameter of the first function call (*block 303 of FIG. 4; p. 18, lines 21-22*) by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device (*p. 4, lines 7-15*). Each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to position a cursor on the first function call displayed in the source code in response to user input. (*P. 23, lines 17-22; FIG. 14*).

The program instructions are further executable to: in response to user input requesting to select a parameter value (*p. 23, lines 9-10*), determine that the cursor is positioned on the first function call (*p. 23, lines 17-22; FIG. 14*) and display a graphical user interface for selecting a parameter value for the first parameter of the first function call (*block 305 of FIG. 4; p. 21, line 10*). The graphical user interface visually indicates the one or more valid parameter values. (*FIG. 7; p. 21, lines 11-13*).

The program instructions are further executable to receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values. (*Block 307 of FIG. 4; p. 22, lines 1-2*). The first parameter value represents a first hardware resource of the measurement device. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value. (*Block 309 of FIG. 4; p. 22, lines 8-18*). Automatically including the

first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device. (*P. 22, lines 11-14; p. 26, lines 15-16*).

Claim 24 recites a computer-readable memory medium (*Block 166 of FIG. 3; p. 8, lines 15-27*) storing program instructions. The program instructions are executable to: in source code of a software program, display a first method call written in a text-based programming language that can be compiled into executable code, wherein the first method call takes a first parameter. (*See block 301 of FIG. 4; FIG. 14; p. 22, lines 17-18*).

The program instructions are further executable to programmatically determine one or more valid parameter values for the first parameter of the first method call (*block 303 of FIG. 4; p. 18, lines 21-22*) by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device. (*p. 4, lines 7-15*) Each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to position a cursor on the first method call displayed in the source code in response to user input. (*P. 23, lines 17-22; FIG. 14*).

The program instructions are further executable to: in response to user input requesting to select a parameter value (*p. 23, lines 9-10*), determine that the cursor is positioned on the first method call (*p. 23, lines 17-22; FIG. 14*) and display a graphical user interface for selecting a parameter value for the first parameter of the first method call (*block 305 of FIG. 4; p. 21, line 10*). The graphical user interface visually indicates the one or more valid parameter values. (*FIG. 7; p. 21, lines 11-13*).

The program instructions are further executable to receive user input to the graphical user interface to select a first parameter value from the one or more parameter values. (*Block 307 of FIG. 4; p. 22, lines 1-2*). The first parameter value represents a first hardware resource of the measurement device. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to automatically modify the first function call displayed in the source code of the software program by including the first

parameter value in the first method call in response to the user input selecting the first parameter value. (*Block 309 of FIG. 4; p. 22, lines 8-18*). Automatically including the first parameter value in the first method call aids a user in modifying the first method call to reference the first hardware resource of the measurement device. (*P. 22, lines 11-14; p. 26, lines 15-16*).

Claim 26 recites a system comprising: one or more processors (*block 160 of FIG. 3; p. 17, lines 3-4*); a display device (*FIG. 1; p. 3, lines 4-6*); a measurement device (*FIG. 2; Blocks 122, 114 and 190 of FIG. 3; p. 12, lines 6-24*); and memory storing program instructions (*block 160 of FIG. 3; p. 17, lines 3-4*). The program instructions stored in the memory are executable by the one or more processors to: in source code of a software program, display on the display device a first function call written in a text-based programming language that can be compiled into executable code. The first function call takes a first parameter. (*See block 301 of FIG. 4; FIG. 14; p. 22, lines 17-18*).

The program instructions stored in the memory are further executable by the one or more processors to programmatically determine one or more valid parameter values for the first parameter of the first function call (*block 303 of FIG. 4; p. 18, lines 21-22*) by invoking software for the measurement device in order to determine one or more hardware resources of the measurement device (*p. 4, lines 7-15*). Each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources. (*P. 20, lines 17-18; FIG. 7*).

The program instructions stored in the memory are further executable by the one or more processors to position a cursor on the first function call displayed in the source code in response to user input. (*P. 23, lines 17-22; FIG. 14*).

The program instructions stored in the memory are further executable by the one or more processors to: in response to user input requesting to select a parameter value (*p. 23, lines 9-10*), determine that the cursor is positioned on the first function call (*p. 23, lines 17-22; FIG. 14*) and display a graphical user interface for selecting a parameter value for the first parameter of the first function call (*block 305 of FIG. 4; p. 21, line 10*). The graphical user interface visually indicates the one or more valid parameter values. (*FIG. 7; p. 21, lines 11-13*).

The program instructions stored in the memory are further executable by the one or more processors to receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values. (*Block 307 of FIG. 4; p. 22, lines 1-2*). The first parameter value represents a first hardware resource of the measurement device. (*P. 20, lines 17-18; FIG. 7*).

The program instructions stored in the memory are further executable by the one or more processors to automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value. (*Block 309 of FIG. 4; p. 22, lines 8-18*). Automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device. (*P. 22, lines 11-14; p. 26, lines 15-16*).

Claim 27 recites a computer-implemented method comprising: in source code of a software program, displaying a first function call written in a text-based programming language that can be compiled into executable code. The first function call takes a first parameter. (*See block 301 of FIG. 4; FIG. 14; p. 22, lines 17-18*).

The method further comprises programmatically determining one or more valid parameter values for the first parameter of the first function call (*block 303 of FIG. 4; p. 18, lines 21-22*) by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device (*p. 4, lines 7-15*). Each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources. (*P. 20, lines 17-18; FIG. 7*).

The method further comprises positioning a cursor on the first function call displayed in the source code in response to user input. (*P. 23, lines 17-22; FIG. 14*).

The method further comprises: in response to user input requesting to select a parameter value (*p. 23, lines 9-10*), determining that the cursor is positioned on the first function call (*p. 23, lines 17-22; FIG. 14*) and displaying a graphical user interface for selecting a parameter value for the first parameter of the first function call (*block 305 of FIG. 4; p. 21, line 10*). The graphical user interface visually indicates the one or more valid parameter values. (*FIG. 7; p. 21, lines 11-13*).

The method further comprises receiving user input to the graphical user interface to select a first parameter value from the one or more valid parameter values. (*Block 307 of FIG. 4; p. 22, lines 1-2*). The first parameter value represents a first hardware resource of the measurement device. (*P. 20, lines 17-18; FIG. 7*).

The method further comprises automatically modifying the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value. (*Block 309 of FIG. 4; p. 22, lines 8-18*). Automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device. (*P. 22, lines 11-14; p. 26, lines 15-16*).

Claim 31 recites a computer-readable memory medium (*Block 166 of FIG. 3; p. 8, lines 15-27*) storing program instructions executable to: display a block diagram of a graphical program, where the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program. The block diagram can be compiled into executable code. (*Block 301 of FIG. 4; p. 10, line 11 – p. 11, line 17*). The plurality of interconnected nodes includes a first node that takes a first input parameter. (*p. 22, lines 19-21*).

The program instructions are further executable to programmatically determine one or more valid parameter values for the first input parameter of the first node (*block 303 of FIG. 4; p. 18, lines 21-22*) by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device (*p. 4, lines 7-15*). Each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to display a graphical user interface for selecting a parameter value for the first input parameter of the first node. (*Block 305 of FIG. 4; p. 21, lines 10-13; p. 22, lines 19-21*). The graphical user interface for selecting the parameter value visually indicates the one or more valid parameter values. (*FIG. 7; p. 21, lines 11-13*).

The program instructions are further executable to receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values. (*Block 307 of FIG. 4; p. 22, lines 1-2*). The first parameter value represents a first hardware resource of the measurement device. (*P. 20, lines 17-18; FIG. 7*).

The program instructions are further executable to automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value. (*Block 309 of FIG. 4; p. 22, lines 20-21*). Automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input. (*P. 23, lines 4-8*).

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Section 103 Rejections (Shulman in view of Sojoodi)

Claims 1, 9, 11, 13, 20-24, 26, 27, 31 and 32 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,026,233 to Shulman et al. (hereinafter “Shulman”) in view of U.S. Patent No. 5,784,275 to Sojoodi et al. (hereinafter “Sojoodi”).

Section 103 Rejection (Shulman in view of Sojoodi and Austin)

Claim 12 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Shulman in view of Sojoodi, and further in view of Austin, U.S. Patent No. 6,370,569 (hereinafter “Austin”).

Section 103 Rejections (Shulman in view of Sojoodi and Molinari)

Claims 33 and 34 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Shulman in view of Sojoodi, and further in view of Molinari et al., U.S. Patent Application Pub. No. 2003/0058280 (hereinafter “Molinari”).

VII. ARGUMENT

Section 103 Rejections (Shulman in view of Sojoodi)

Claims 1, 9, 11, 13, 20-24, 26, 27, 31 and 32 stand rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,026,233 to Shulman et al. (hereinafter “Shulman”) in view of U.S. Patent No. 5,784,275 to Sojoodi et al. (hereinafter “Sojoodi”). Appellant respectfully traverses this rejection.

Independent Claims 1, 24, 26 and 27

Claim 1 recites in pertinent part:

programmatically determine one or more valid parameter values for the first parameter of the first function call by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources; (*emphasis added*)

Appellant respectfully submits that Shulman and Sojoodi, taken either singly or in combination, do not teach these limitations in combination with the other limitations recited in claim 1.

With respect to the limitation of invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, the Examiner cites Sojoodi at Col. 7, lines 3-17. Here Sojoodi teaches querying an object manager for a list of VISA (Virtual Instrument Software Architecture) classes associated with an instrument. The VISA classes of the instrument correspond to the possible VISA interface types of the instrument. On p. 2 of the Office Action the Examiner states that:

“querying the object manager for a list of classes of the instrument, where the classes correspond to possible VISA interface types of the instrument” (see, for example, column 7, lines 3-17) represents querying the object manager for a list of one or more hardware resources of the instrument.

Appellant respectfully submits that the VISA classes determined by querying the object manager are not hardware resources of a device, as recited in claim 1. Sojoodi teaches:

A new instrumentation programming environment is referred to as the Virtual Instrument Software Architecture (VISA). The VISA architecture provides an object-oriented environment for controlling instrumentation systems and for providing a user or developer with the capability to develop instrument drivers and application software for controlling instrumentation systems. The VISA development system is independent of I/O interface type, operating system, and programming language while also providing a consistent Application Programming Interface (API) to the user. (*Col. 3, line 64 – Col. 4, line 6*)

Thus, VISA is an object-oriented programming environment. Appellant respectfully submits that the classes determined by querying the object manager are object-oriented software classes. Sojoodi also teaches:

The VISA library 52 comprises executable functions which are called by the VI 50 to perform various operations in order to control the instrument 54. Examples of the executable functions are viOpen(), viClose(), viRead(), viwrite(), viSetAttribute(), viGetAttribute(), viPeek(), viPoke(), viAssertTrigger(), viClear(), and viReadSTB(). These functions perform operations to control the instrument 54 as defined in the VISA specification in Appendix C. VISA resources comprise the operations, along with numerous attributes defined in the VISA specification. The VISA nodes 66 and VISA session control 74 comprise various classes and objects, according to the notion of classes and objects in the art of object-oriented programming. These classes and objects correspond to the resources in the VISA I/O Control library 52 as will be explained in more detail below. (*Col. 12, lines 49-64*)

and

The graphical programming environment of the present invention comprises an object manager. The object manager is a central repository of class information. (*Col. 24, lines 29-31*)

Thus, it is clear that the classes in the list of classes returned by the object manager are object-oriented software classes. A software class is not at all the same thing as a hardware resource. VISA defines an object-oriented style interface for communicating with instruments in an interface-independent manner, and a VISA class is simply a grouping of particular VISA operations. Appellant thus respectfully submits that the Examiner's interpretation of a VISA class as a hardware resource is erroneous.

Appellant notes that claim 1 further recites:

receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device. (*emphasis added*)

As noted above, the Examiner has interpreted the recited “hardware resources” as the list of classes returned by the object manager. Claim 1 recites that “each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources”. Thus, under the Examiner’s interpretation, each of the one or more valid parameter values represents a respective class of the list of classes. Claim 1 also recites, “automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call”. Thus, under the Examiner’s interpretation the first function call is modified by including a first parameter value representing a particular VISA class in the first function call. This means that the particular VISA class would be passed as a parameter to the first function call. Appellant notes that in object-oriented programming, you typically do not pass a class into a function call as a parameter. In fact, Appellant is not aware of any object-oriented programming languages which even allow classes to be passed as parameters into function calls. A class contains various methods or functions which can be invoked. Thus, one might invoke a function contained in a class by passing various parameters into that function, but it is not at all clear why one would want to pass a class as a parameter into a function call (or even whether this is syntactically allowable).

Appellant thus respectfully submits that independent claim 1 is patentably distinct over the cited art for at least the reasons set forth above.

Inasmuch as the independent claims 26 and 27 recite limitations similar or identical to those discussed above with reference to claim 1, Appellant respectfully submits that these claims are also patentably distinct over the cited art.

The independent claim 24 also recites similar limitations as claim 1, except that instead of a first function call it recites a first method call. The claim recites in pertinent part:

programmatically determine one or more valid parameter values for the first parameter of the first method call by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

As discussed above with respect to claim 1, Appellant respectfully submits that the Examiner's equation of software classes with hardware resources is erroneous. Appellant thus respectfully submits that claim 24 is also patentably distinct over the cited art.

Independent Claim 31

Claim 31 recites in pertinent part:

programmatically determine one or more valid parameter values for the first input parameter of the first node by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

As discussed above with respect to claim 1, Appellant respectfully submits that the Examiner's equation of software classes with hardware resources is erroneous. Appellant thus respectfully submits that claim 31 is patentably distinct over the cited art for at least this first reason.

Furthermore, claim 31 also recites:

display a block diagram of a graphical program, wherein the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein the block diagram can be compiled into executable code, wherein the plurality of interconnected nodes includes a first node that takes a first input parameter;

and

automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input. (Emphasis added)

Appellant notes that these limitations are distinct from those of claims 1, 24, 26, and 27. Whereas claims 1, 24, 26, and 27 relate to modifying text-based source code by automatically modifying a first function call (or method call) written in a text-based programming language, claim 31 relates to modifying a graphical program block diagram which includes a plurality of interconnected nodes by automatically configuring a first node of the plurality of interconnected nodes with the selected first parameter value.

Appellant respectfully submits that Shulman and Sojoodi, taken either singly or in combination, do not teach the recited limitation of automatically configuring the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input.

With respect to this limitation the Examiner cites Shulman's teaching of modifying a function call written in a text-based programming language. However, modifying a function call written in a text-based programming language is not the same as automatically configuring a node in a block diagram of a graphical program with a parameter value.

Furthermore, Appellant respectfully submits that Sojoodi does not remedy this deficiency in Shulman's teaching. Appellant respectfully submits that Sojoodi, taken either singly or in combination with Shulman, does not teach "automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input".

Claim 32

Claim 32 recites the further limitations of:

wherein automatically configuring the first node with the first parameter value comprises automatically wiring the first parameter value to an input terminal of the first node;

wherein updating the block diagram comprises displaying a wire connecting the first parameter value to the input terminal of the first node.

The Examiner states that:

Specifically, Sojoodi describes that configuring the block diagram comprises wiring a parameter value to an input terminal of a node and displaying the wire (see, for example, column 5, lines 52-67).

The cited section of Sojoodi teaches:

The VISA node icons in the block diagram comprise terminals which the user wires together using a wiring tool.

Thus, Sojoodi teaches that the user manually wires the nodes together using a wiring tool. Appellant respectfully submits that Sojoodi, taken either singly or in combination with Shulman, does not teach automatically wiring the first parameter value to an input terminal of the first node, as recited in claim 32.

Appellant thus respectfully submits that claim 32 is separately patentable over the cited art.

Claims 20 and 21

Claim 20 recites the further limitations of: “wherein the source code is displayed in a first window,” and “wherein said displaying the graphical user interface comprises displaying the graphical user interface in a separate window apart from the first window”.

Claim 21 recites the further limitations of: “wherein the source code is displayed in a first portion of a first window,” and “wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the first window”.

With respect to both of these claims, the Examiner cites FIG. 8. In the rejection of claim 20, the Examiner states:

see, for example, FIG. 8, which shows that the graphical user interface 850 is displayed in a separate window

However, in the rejection of claim 21, the Examiner states:
see, for example, FIG. 8, which shows that the graphical user interface 850 is displayed in a second portion of the first window

These statements are inconsistent with each other. They cannot both be true. In FIG. 8, the graphical user interface 850 is only displayed in one window. The window in which it is displayed is either the same first window in which the source code is displayed, or it is a separate window apart from the first window. The window in which the graphical user interface 850 is displayed in FIG. 8 cannot be both the first window and another window separate from the first window.

Appellant notes that Shulman teaches that the graphical user interface 850 is an assist window. (See Col. 12, lines 25-36). Appellant respectfully submits that the assist window is a separate window from the window in which Shulman's source code is displayed. Appellant thus respectfully submits that Shulman does not teach the limitation in claim 21 of "wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the first window". Accordingly Appellant respectfully submits that claim 21 is separately patentable over the cited art.

Section 103 Rejections (Shulman in view of Sojoodi and Molinari)

Claims 33 and 34 stand rejected under 35 U.S.C. 103(a) as being unpatentable over Shulman in view of Sojoodi, and further in view of Molinari et al., U.S. Patent Application Pub. No. 2003/0058280 (hereinafter "Molinari"). Appellant respectfully traverses this rejection.

Claim 33

Claim 33 recites the further limitations of, "wherein the measurement device includes a plurality of channels" and "wherein invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device comprises invoking the software for the measurement device in order to determine the plurality of channels."

In the rejection of claim 33 the Examiner states:

Therefore, it would have been obvious to those of ordinary skill in the art at the time the invention was made to implement the teachings of Shulman and Sojoodi such that the measurement device includes a plurality of channels and that invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device comprises invoking the software for the measurement device in order to determine the plurality of channels.

Appellant respectfully disagrees. As discussed above with respect to claim 1, the Examiner has interpreted the limitation of “invoking software for a measurement device” with Sojoodi’s teaching of querying the object manager. The object manager is a central repository of class information (see Col. 24, lines 29-31), and the purpose of querying the object manager is to receive a list of classes. The Examiner is proposing that Sojoodi’s invention be modified so that instead of returning a list of classes, the object manager would instead return the plurality of channels of the measurement device. Appellant respectfully submits that this would destroy a key principle of Sojoodi’s invention: namely, the use of the object manager as a central repository of class information. If the object manager were modified to return a list of channels instead of a list of classes then the object manager could no longer fulfill its intended purpose as a central repository of class information.

Appellant thus respectfully submits that it would not be obvious to modify Sojoodi’s teaching as proposed by the Examiner because this would violate a key principle of Sojoodi’s invention. Appellant thus respectfully submits that claim 33 is separately patentable over the cited art.

Claim 34

Claim 34 recites the further limitation of, “wherein said invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device is performed in response to the user input requesting to select a parameter value.” In the rejection of claim 34 the Examiner states:

With respect to claim 34 (new), the rejection of claim 1 is incorporated. Shulman in view of Sojoodi does not explicitly describe that said invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device is performed in response to the user input requesting to select a parameter value.

Nonetheless, in an analogous art, Molinari teaches invoking software for a measurement device in order to determine one or more hardware resources of the measurement device (see, for example, paragraph [0150]). The invoking is performed in response to user input requesting to select a data source (see, for example, paragraph [0149]).

At paragraphs 0149-150 Molinari teaches:

[0149] ... Upon the selection of a data source by the user, from the listing in said combo box, data sink aspect 130 establishes a data link 138 to the selected data source aspect.

[0150]Picking up in FIG. 12 the continuation of the above-described process, data sink aspect 130, upon establishing a data link with the selected data source aspect 140, requests said data source aspect to provide the number of ("DAQ") data channels it provides and the basic properties of each of those channels, including its name, state and ID.

Thus, Molinari teaches that the user first selects a data source, and then the data link to the selected data source aspect is established, and the data source aspect is invoked to obtain the channel information. Thus, the invoking is performed after the data source has already been selected. The selected data source equates to the selected first parameter value recited in claim 1. However, claims 1 and 34 require that the software for the measurement device be invoked in order to determine the one or more hardware resources before the first parameter value, not after as taught by Molinari. In claim 1 and 34, the graphical user interface visually indicates the one or more valid parameter values, and thus, the one or more hardware resources must be determined before they can be displayed in the graphical user interface.

The Examiner has apparently interpreted the recited "user input requesting to select a parameter value" as Molinari's user input which actually selects a particular data source from a list. Appellant respectfully submits that this interpretation is erroneous. Under this interpretation, the "user input requesting to select a parameter value" clearly cannot occur until after the graphical user interface listing the valid parameter values (e.g., data sources) has already been displayed. (This is because the user selects the data source from the graphical user interface or list, and thus the list must be displayed first before the user can select an item from the list.)

The Examiner's interpretation is clearly erroneous because claims 1 and 34 require that the "user input requesting to select a parameter value" occurs first before the graphical user interface or list is displayed. (See claim 1 - the graphical user interface which visually indicates the one or more valid parameter values is displayed in response to the user input requesting to select a parameter value.) Thus, the Examiner's interpretation of the "user input requesting to select a parameter value" as the user input which actually selects the particular parameter value or data source from the displayed graphical user interface or list is erroneous. Instead, the "user input requesting to select a parameter value" is the user input that causes the graphical user interface or list to be displayed.

Appellant thus respectfully submits that Molinari, taken either singly or in combination with Shulman and Sojoodi, does not teach "wherein said invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device is performed in response to the user input requesting to select a parameter value." Accordingly, Appellant respectfully submits that claim 34 is separately patentable over the cited art.

Section 103 Rejections (Shulman in view of Sojoodi and Austin)

Claim 12 stands rejected under 35 U.S.C. 103(a) as being unpatentable over Shulman in view of Sojoodi, and further in view of Austin, U.S. Patent No. 6,370,569 (hereinafter "Austin").

Claim 12 depends on claim 1, which Appellant argued above is patentable over the cited art. Appellant thus respectfully submits that claim 12 is also patentably distinct by virtue of being depending upon a patentable base claim.

VIII. CONCLUSION

For the foregoing reasons, it is submitted that the Examiner's rejection of the claims was erroneous, and reversal of the decision is respectfully requested.

The fee of \$540.00 for filing this Appeal Brief is being paid concurrently via EFS-Web. If any extensions of time (under 37 C.F.R. § 1.136) are necessary to prevent the above-referenced application(s) from becoming abandoned, Applicant(s) hereby petition for such extensions. The Commissioner is hereby authorized to charge any fees which may be required or credit any overpayment to Meyertons, Hood, Kivlin, Kowert & Goetzl P.C., Deposit Account No. 50-1505/5150-77600/JCH.

Respectfully submitted,

/Jeffrey C. Hood/

Jeffrey C. Hood, Reg. #35198

ATTORNEY FOR APPLICANT(S)

Meyertons Hood Kivlin Kowert & Goetzl, P.C.
P.O. Box 398
Austin, TX 78767-0398
Phone: (512) 853-8800

Date: 2009-08-26 JCH/JLB

IX. CLAIMS APPENDIX

The following lists the claims as incorporating entered amendments, and as on appeal.

1. (Previously Presented) A computer-readable memory medium storing program instructions executable to:

in source code of a software program, display a first function call written in a text-based programming language that can be compiled into executable code, wherein the first function call takes a first parameter;

programmatically determine one or more valid parameter values for the first parameter of the first function call by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

position a cursor on the first function call displayed in the source code in response to user input;

in response to user input requesting to select a parameter value, determine that the cursor is positioned on the first function call and display a graphical user interface for selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface visually indicates the one or more valid parameter values;

receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device.

2-8. (Canceled)

9. (Previously Presented) The computer-readable memory medium of claim 1,
wherein the measurement device comprises a GPIB device;
wherein said determining the one or more hardware resources of the measurement device comprises determining one or more hardware resources of the GPIB device;
wherein the first parameter value represents a first hardware resource of the GPIB device;
wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first hardware resource of the GPIB device.

10. (Canceled)

11. (Previously Presented) The computer-readable memory medium of claim 1,
wherein the measurement device comprises a DAQ device;
wherein said determining the one or more hardware resources of the measurement device comprises determining one or more hardware resources of the DAQ device;
wherein the first parameter value represents a first hardware resource of the DAQ device;
wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first hardware resource of the DAQ device.

12. (Previously Presented) The computer-readable memory medium of claim 1,
wherein said determining the one or more valid parameter values comprises determining one or more universal resource locators (URLs) that represent the one or more hardware resources of the measurement device;
wherein the first parameter value comprises a first URL of the one or more URLs;

wherein said automatically including the first parameter value in the first function call comprises automatically configuring the first function call with a reference to the first URL.

13. (Previously Presented) The computer-readable memory medium of claim 1, wherein the program instructions are further executable to:

receive user input specifying filtering criteria for the parameter values;
wherein the graphical user interface visually indicates only a subset of the valid parameter values, wherein the subset is determined based on the specified filtering criteria.

14-19. (Canceled)

20. (Previously Presented) The computer-readable memory medium of claim 1,

wherein the source code is displayed in a first window;
wherein said displaying the graphical user interface comprises displaying the graphical user interface in a separate window apart from the first window.

21. (Previously Presented) The computer-readable memory medium of claim 1,

wherein the source code is displayed in a first portion of a first window;
wherein said displaying the graphical user interface comprises displaying the graphical user interface in a second portion of the first window.

22. (Previously Presented) The computer-readable memory medium of claim 1,

wherein the graphical user interface displays the one or more valid parameter values as a list;

wherein said receiving user input to the graphical user interface to select the first parameter value comprises receiving user input to the graphical user interface to select the first parameter value from the list.

23. (Previously Presented) The computer-readable memory medium of claim 1,

wherein said programmatically determining the one or more valid parameter values includes programmatically determining one or more property values;

wherein said receiving user input to the graphical user interface to select the first parameter value comprises receiving user input to the graphical user interface to select a first property value;

wherein the first property value is automatically included in the first function call in response to the user input selecting the first property value.

24. (Previously Presented) A computer-readable memory medium storing program instructions executable to:

in source code of a software program, display a first method call written in a text-based programming language that can be compiled into executable code, wherein the first method call takes a first parameter;

programmatically determine one or more valid parameter values for the first parameter of the first method call by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

position a cursor on the first method call displayed in the source code in response to user input;

in response to user input requesting to select a parameter value, determine that the cursor is positioned on the first method call and display a graphical user interface for selecting a parameter value for the first parameter of the first method call, wherein the graphical user interface visually indicates the one or more valid parameter values;

receive user input to the graphical user interface to select a first parameter value from the one or more parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first method call in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first method call aids a user in modifying the first method call to reference the first hardware resource of the measurement device.

25. (Canceled)

26. (Previously Presented) A system comprising:

one or more processors;

a display device;

a measurement device; and

memory storing program instructions;

wherein the program instructions stored in the memory are executable by the one or more processors to:

in source code of a software program, display on the display device a first function call written in a text-based programming language that can be compiled into executable code, wherein the first function call takes a first parameter;

programmatically determine one or more valid parameter values for the first parameter of the first function call by invoking software for the measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

position a cursor on the first function call displayed in the source code in response to user input;

in response to user input requesting to select a parameter value, determine that the cursor is positioned on the first function call and display a graphical user interface for

selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface visually indicates the one or more valid parameter values;

receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device.

27. (Previously Presented) A computer-implemented method comprising:

in source code of a software program, displaying a first function call written in a text-based programming language that can be compiled into executable code, wherein the first function call takes a first parameter;

programmatically determining one or more valid parameter values for the first parameter of the first function call by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

positioning a cursor on the first function call displayed in the source code in response to user input;

in response to user input requesting to select a parameter value, determining that the cursor is positioned on the first function call and displaying a graphical user interface for selecting a parameter value for the first parameter of the first function call, wherein the graphical user interface visually indicates the one or more valid parameter values;

receiving user input to the graphical user interface to select a first parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically modify the first function call displayed in the source code of the software program by including the first parameter value in the first function call in

response to the user input selecting the first parameter value, wherein automatically including the first parameter value in the first function call aids a user in modifying the first function call to reference the first hardware resource of the measurement device.

28-30. (Canceled)

31. (Previously Presented) A computer-readable memory medium storing program instructions executable to:

display a block diagram of a graphical program, wherein the block diagram includes a plurality of interconnected nodes visually indicating functionality of the graphical program, wherein the block diagram can be compiled into executable code, wherein the plurality of interconnected nodes includes a first node that takes a first input parameter;

programmatically determine one or more valid parameter values for the first input parameter of the first node by invoking software for a measurement device in order to determine one or more hardware resources of the measurement device, wherein each of the one or more valid parameter values represents a respective hardware resource of the one or more hardware resources;

display a graphical user interface for selecting a parameter value for the first input parameter of the first node, wherein the graphical user interface for selecting the parameter value visually indicates the one or more valid parameter values;

receive user input to the graphical user interface to select a first parameter value from the one or more valid parameter values, wherein the first parameter value represents a first hardware resource of the measurement device; and

automatically configure the first node with the first parameter value in response to the user input selecting the first parameter value, wherein automatically configuring the first node with the first parameter value comprises automatically updating the displayed block diagram to visually indicate that the first node receives the first parameter value as input.

32. (Previously Presented) The computer-readable memory medium of claim 31,

wherein automatically configuring the first node with the first parameter value comprises automatically wiring the first parameter value to an input terminal of the first node;

wherein updating the block diagram comprises displaying a wire connecting the first parameter value to the input terminal of the first node.

33. (Previously Presented) The computer-readable memory medium of claim 1,

wherein the measurement device includes a plurality of channels;

wherein invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device comprises invoking the software for the measurement device in order to determine the plurality of channels.

34. (Previously Presented) The computer-readable memory medium of claim 1,

wherein said invoking the software for the measurement device in order to determine the one or more hardware resources of the measurement device is performed in response to the user input requesting to select a parameter value.

X. EVIDENCE APPENDIX

No evidence submitted under 37 CFR §§ 1.130, 1.131 or 1.132 or otherwise entered by the Examiner is relied upon in this appeal.

XI. RELATED PROCEEDINGS APPENDIX

There are no related proceedings.